IN THE SPECIFICATION

Please amend the specification as follows:

Replace the paragraph spanning pages 1-2 between page 1, line 20, and page 2, line 8 of the specification with the following:

A well-known example of a Feistel cipher is DES, consisting of sixteen rounds. In each round, first the 32 bits of the right half of the data are expanded to 48 bits. Next, ~~an~~ a 48 bit round key, which is computed from a 56 bit DES key with a key scheduling algorithm, is bit-wise added modulo two to these 48 bits. Then a layer of S-boxes performs a non-linear operation on the data. In DES, the S-box layer consist of eight six-to-four bit S-boxes in parallel, i.e. each of the S-boxes converts a 6-bit input block into a 4-bit output block using one fixed mapping table per S-box. The output of the S-box layer is a 32 bit data block. The linear transformation, which is performed on this 32 bit data block, is a bit-permutation, which ensures that bit changes caused by an S-box are propagated over many other ones in the following round(s). A drawback of DES is its small key size of 56 bits, which is

considered to be insufficient nowadays for offering a high level of security. However, an exhaustive key search can be avoided by using a longer key combined with a different key scheduling algorithm for computing the sixteen 48-bit round keys. The two most powerful attacks on DES published in the open literature are differential and linear cryptanalysis, which are general attacks that can be applied to a wide range of block ciphers. It has been shown that DES can not be strengthened much against these attacks by modifying the key length and/or the key-scheduling algorithm. However, changes in the round function of the algorithm can influence its strength against these attacks considerably.

Replace the paragraph on page 3, between lines 1-2 of the specification with the following:

~~As defined in the measure of the dependent claim 2, the~~ The new columns can be (pseudo-)randomly generated in order to find suitable columns.

Replace the paragraph on page 3, between lines 3-10 of the specification with the following:

~~As defined in the measure of the dependent claim 3, the~~ The resulting matrix C is permuted to find a linear transformation matrix with the associated linear error-correcting code having a predetermined multi-bit weight. ~~As defined in the measure of the dependent claim 4, this~~ This multi-bit weight ensures proper diffusion over the S-boxes of the cipher. For instance, for an S-box layer consisting of a number of S-boxes operating in parallel, in which each S-box provides an m-bit output, it is relevant to look at the diffusion of m-bit parts of the words in the associated binary error-correcting code, which can be expressed in the minimum m-bit weight over all non-zero codewords.

Replace the paragraph on page 10, between lines 3-16 of the specification with the following:

Due to the construction of the round function with the multi-bit S-boxes, also good diffusion properties on this multi-bit level are desirable. For four-bit S-boxes, this can be expressed as follows (variations for other number of bits fall well within the skills of persons skilled in the art). If the 4-bit vectors n_i (i = 0,1, ... ,7) of a codeword $c \in Z_2^{32}$ are defined as $c =: (n_0 \,||\, n_1 \,||$

... || n_7) then the nibble weight of c is defined as NW(c) := #{ i || $n_i \neq (0,0,0,0)$, i = 0,1, ... ,7)}. The diffusion properties on nibble-level can be expressed in terms of the minimum nibble weight over all non-zero codewords; the higher this minimum weight, the better the diffusion properties on nibble-level. To achieve a high diffusion at multi-bit level (in the example, at nibble level), in step 530, two permutation matrices $P_1, P_2 \in Z_2^{32\times32}$ are selected (in step 532) such that all codewords in the [64,32,12] code with generator matrix (I || P_1 C P_2) have a high nibble weight, as verified in step 534. The finally found matrix A := P_1 C P_2 is used for the linear transformation. In a preferred embodiment, the permutation matrices P_1 and P_2 are (pseudo-) randomly generated. It can be verified in step 536 that the minimum nibble weight of the code generated by (I || A) equals seven.